Anton Tyurin
Head of Attack Detection Team

Detect Malicious Communications Even Under TLS
Whoami

Responsible for network attacks detection

Doing threat hunting in clients’ infrastructure

Developing Suricata rules: more than 4000

twitter.com/AttackDetection
Malware Evades IDS Using Encryption

Remcos (TCP)

Multiple Malware Campaigns Distributing Remcos RAT Via Malicious Excel and Word Documents

An Italian malware developer by the name of Viotto has published his latest creation, the Remcos RAT (Remote Access Trojan), which he’s selling on underground hacking forums for a price that varies between $58 and $589, payable in various anonymous digital currencies.

gbhackers.com/multiple-malware-campaigns-remcos-rat/

Adwind/AlienSpy (TLS)

Adwind Trojan circumvents antivirus software to infect your PC

A spam campaign spreading the RAT uses a number of tricks to fool signature-based antivirus solutions.

zethe.com/article/this-is-how-the-adwind-trojan-tricks-antivirus-software-to-infect-your-pc/

From a fake wallet to a Java RAT

Posted: January 18, 2017 by Malwarebytes Labs
Last updated: January 23, 2017

This malware came in a phishing e-mail – disguised as a Bitcoin wallet. After clicking the link, user receives a JAR file (hosted at Dropbox): wallet.aes.json.jar, that turns out to be a RAT – Adwind.

blog.malwarebytes.com/cybercrime/2017/01/from-a-fake-wallet-to-a-java-rat/
Typical Reaction to Encrypted Traffic

Denial

Anger

Bargaining

Depression

Acceptance
How Stream Works

Client

Packet Data Length, B

Packet #1

Packet #2

Packet #3

Packet #4

Packet #5

Packet #6

Server

Stream, B

Packet #1

Packet #2

Packet #3

Packet #4

Packet #5

Stream, B
When There Is More Than One Sample
When There Is More Than One Sample

Packet Data Length, B

Client Stream, B
The More areas, The More Path, The less mess
The More Areas, the More Path, the Less Mess
TCP Rules and Keywords

Rule Example

```
alert tcp $HOME_NET any -> $EXTERNAL_NET any
  (msg:"pkt checker 0";
   flow:established, to_server;
   dsize:199<>601;
   stream_size:client,>1024;
   stream_size:client, <2049;
   sid: 1; rev: 1;)
```
Rules, Streams and Flowbits

Flowbits
Rule 1  ON
Rule 2  OFF
Rule 3  OFF
Rule 4  OFF
Rule 5  OFF
1 stream – 1 alert

**Flowbits**

<table>
<thead>
<tr>
<th>Rule 1</th>
<th>Rule 2</th>
<th>Rule 3</th>
<th>Rule 4</th>
<th>Rule 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>OFF</td>
<td>ON</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
</tr>
<tr>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>OFF</td>
<td>ALERT</td>
</tr>
</tbody>
</table>
REMCOS Communication

Server
- 32 - c
- 84 - i
- 70 - i
- 88 - i

Client
- 748 - sysinfo
- 84 - i
- 70 - i
- 88 - i

Stream, B

c - command

i - information

d_size

[DataStart]...addnew|cmd|BitTarget|cmd|
[DataStart]...ping|cmd|0|cmd|20
[DataStart]E...pong|cmd|0|cmd|P.r.o.g.r.a.m
[DataStart]...ping|cmd|0|cmd|20
[DataStart]E...pong|cmd|0|cmd|P.r.o.g.r.a.m.
[DataStart]...ping|cmd|0|cmd|20
[DataStart]7...pong|cmd|0|cmd|S.t.a.r.t.|cmd
[DataStart]...ping|cmd|0|cmd|20
[DataStart]7...pong|cmd|0|cmd|S.t.a.r.t.|cmd
[DataStart]...ping|cmd|0|cmd|20
[DataStart]...pong|cmd|0|cmd|T.a.s.k. . S.w.
[DataStart]...ping|cmd|0|cmd|20
[DataStart]I...pong|cmd|0|cmd|T.a.s.k. . S.w.
REMHCOS Detection Rules

**Rule 1**
server,=,1;
client,>,200;
client,<,911;
dsize:200<>910;

**Rule 2**
server,>,25;
server,<,35;
client,>,200;
client,<,911;
dsize:25<>35;

**Rule 3**
server,>,25;
server,<,35;
client,>,225;
client,<,1210;
dsize:42<>270;

**Rule 4**
server,>,50;
server,<,200;
client,>,225;
client,<,1210;
dsize:25<>200;

**Rule 5**
server,>,50;
server,<,200;
client,>,246;
client,<,1400;
dsize:44<>270;
**Remcos Detection Rules**

<table>
<thead>
<tr>
<th>Flowbits</th>
<th>Flowbits</th>
<th>Flowbits</th>
<th>Flowbits</th>
<th>Flowbits</th>
</tr>
</thead>
<tbody>
<tr>
<td>flowbits: set, FB_0; flowbits: noalert;</td>
<td>flowbits: isset, FB_0; flowbits: unset, FB_0; flowbits: set, FB_1; flowbits: noalert;</td>
<td>flowbits: isset, FB_1; flowbits: unset, FB_1; flowbits: set, FB_2; flowbits: noalert;</td>
<td>flowbits: isset, FB_2; flowbits: unset, FB_2; flowbits: set, FB_3; flowbits: noalert;</td>
<td>flowbits: isset, FB_3; flowbits: unset, FB_3;</td>
</tr>
</tbody>
</table>
ADWIND Communications

app_data len

Server

| 4 - Magic |
| 4 - Magic |

Client

| ~2500 - System Information |

Command

| 17 |

Command

| 17 |

Command

| 17 |

Response

| 111 |

Response

| 38 |

Response

| 59 |

Command

| 36 |

Response

| 58 |

Response

| 79 |

0x ac ed 00 05
0x ac ed 00 05

{"INSTALL":true,"MODULE_PATH":"P/vpn/rR.En","PLUGIN_FOLDER":"ZNhokvZHHIB","JRE_FOLDER":"xReBRo","SERVER_VERSION":"1.2.0","JAR_EXTENSIONS":"FLnsIkJ","LAST_MODIFIED":1521156818528,"OS_NAME":"Windows"}

{"COMMAND":104}

{"COMMAND":114}

{"COMMAND":115}

{"...Process Explorer - Sysinternals: www.sysinte...}"

{"IDLE":"00 : 00 : 00","COMMAND":4}

{"...","Local Area Connection 4",}

{"PLUGIN_ID":"009","COMMAND":108}

{"ACTIVE_WINDOW":........","COMMAND":5}

{"ACTIVE_WINDOW":........","COMMAND":5}
TLS 1.1, 1.2 Record Layer CBC Block Cipher

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Version</td>
<td>Length</td>
<td>IV</td>
<td>Encrypt</td>
</tr>
<tr>
<td>1 Byte</td>
<td>2 Bytes</td>
<td>2 Bytes</td>
<td>Block size</td>
<td>Data</td>
</tr>
<tr>
<td>5 Bytes</td>
<td>16 Bytes</td>
<td>Block size*N</td>
<td>48[SHA384] +16*N Bytes</td>
<td>Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Pad</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>HMAC</td>
</tr>
</tbody>
</table>
TLS 1.1, 1.2 AES256 CBC SHA384

Quantization
Rounding and truncation are typical examples of quantization process.
# ADWIND Block Size Comparison: Plaintext and TLS

<table>
<thead>
<tr>
<th>App Data Len</th>
<th>Plain</th>
<th>Server</th>
<th>Client</th>
<th>Encrypted</th>
<th>Server</th>
<th>Client</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 - Magic</td>
<td>app_data len</td>
<td>4 - Magic</td>
<td>~2500 - System Information</td>
<td>80 = 4+12+64</td>
<td>80</td>
<td>1104 - max fragment len</td>
</tr>
<tr>
<td>17 - Command</td>
<td>96 = 17+15+64</td>
<td>96 - 15b</td>
<td>1104 - max fragment len</td>
<td>96 - 15b</td>
<td>176 = 111+1+64</td>
<td>112 - 10b</td>
</tr>
<tr>
<td>111 - Response</td>
<td>128 - 5b</td>
<td>112 - 12b</td>
<td>560 - 9b</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
TLS Rules and Keywords

Record Header

- 17 - type is 0x17 (application data)
- 03 03 - protocol version (TLS 1.2)
- 04 50 - the length of the record payload is 0x450 (1104) bytes

Rule Example

```plaintext
alert tls_

flow:established, to_server;
content: "[1703]"; depth:2;
byte_test: 2, >=,900, 1, relative;
byte_test: 2, <=,1104, 1, relative;
stream_size: server, >,1758;
stream_size: server, <,2124;
stream_size: client, >,1376;
stream_size: client, <,5922;
flowbits: isset, FB332502_1;
flowbits: unset, FB332502_1;
flowbits: set, FB332502_2;
flowbits: noalert;
```
Rule 1
server, >1,689;
server, <,2,124;
client, >,447;
client, <,1,722;
content: "[0050]";

Rule 2
server, >,1,658;
server, <,2,124;
client, >,447;
client, <,4,222;
content: "[0050]";

Rule 3
server, >,1,758;
server, <,2,124;
client, >,1,376;
client, <,5,922;
content: "[0060]";
btest: 900<=1104

Rule 4
server, >,1,843;
server, <,2,924;
client, >,1,476;
client, <,8,722;
content: "[0060]";

Rule 5
server, >,1,843;
server, <,3,036;
client, >,1,476;
client, <,8,834;
content: "[0060]";
ADWIND Detection Rules

Flowbits
flowbits: set, FB_0;
flowbits: noalert;

Flowbits
flowbits: isset, FB_0;
flowbits: unset, FB_0;
flowbits: set, FB_1;
flowbits: noalert;

Flowbits
flowbits: isset, FB_1;
flowbits: unset, FB_1;
flowbits: set, FB_2;
flowbits: noalert;

Flowbits
flowbits: isset, FB_2;
flowbits: unset, FB_2;
flowbits: set, FB_3;
flowbits: noalert;

Flowbits
flowbits: isset, FB_3;
flowbits: unset, FB_3;
How to Inspect TLS or to Shoot Yourself in the Foot

```yaml
app-layer:
  protocols:
    krb5:
      enabled: @rust_config_enabled@
    1kev2:
      enabled: yes
    tls:
      enabled: yes
      detection-ports:
        dp: 443

# What to do when the encrypted communications start:
# - full: keep tracking and inspection as normal. Unmodified content
# - keyword signatures are inspected as well.
encrypt-handling: full
```
Suricata Rules for Encrypted Communications

github.com/ptresearch/AttackDetection/tree/master/Suricon2018

1 contributo
Questions?

atyurin@ptsecurity.com
twitter.com/AttackDetection
ptsecurity.com